MICRO-507 | Legged Robots | EPFL
Article Presentation

Joaquin Vasquez-Romo | Paul Huot-Marchand | Josua Bubenzer
Group 12

4 Nov. 2025

parkour in the wild:

Learning a general and extensible agile locomotion policy using multi-expert distillation and RL Fine-tuning

by Nikita Rudin, Junzhe He, Joshua Aurand, and Marco Hutter

01

Starting Point

Legged robots are suitables for **unstructured terrains.**

But **software is limitating performances.**

Current method : train robot for each specific task.

→ **unefficient and redundant**

Idea:

re-use and combine the abilities of individual expert skills into a single, general controller.

02

Result

03

Approach

Expert Skill Training

Train specific skills

Distillation

Merge the skills

RL Fine-tuning

Improve performance

Step 1

Expert Skill Training

Reinforcement Learning

Definition: A skill is the motion required to cross a specific terrain or obstacle

Train specific skills independently using RL

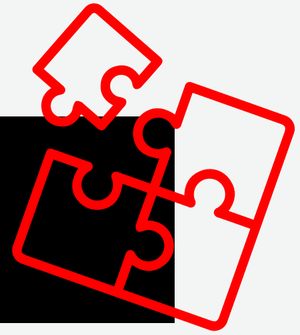
LEARN 9 SKILLS

walking	jumping	walking on stepping stones
climbing	crouching	crossing narrow beams
climbing down	jumping over low walls	climbing piles of boulders

Time consuming but done only once!

Step 2

Distillation



Supervised Learning

Merge the learned skills in one model.

CAPABLE OF:

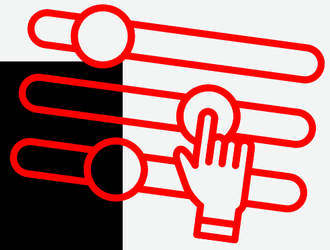
1. **Recognizing** the terrain type.
2. **Selecting** the correct motion from its internal “expert dictionary”.

How ?

This step is done using a **teacher-student** process.

Step 3

RL Fine-tuning



Reinforcement Learning

Improve the Performance and Generalization to unseen terrains.

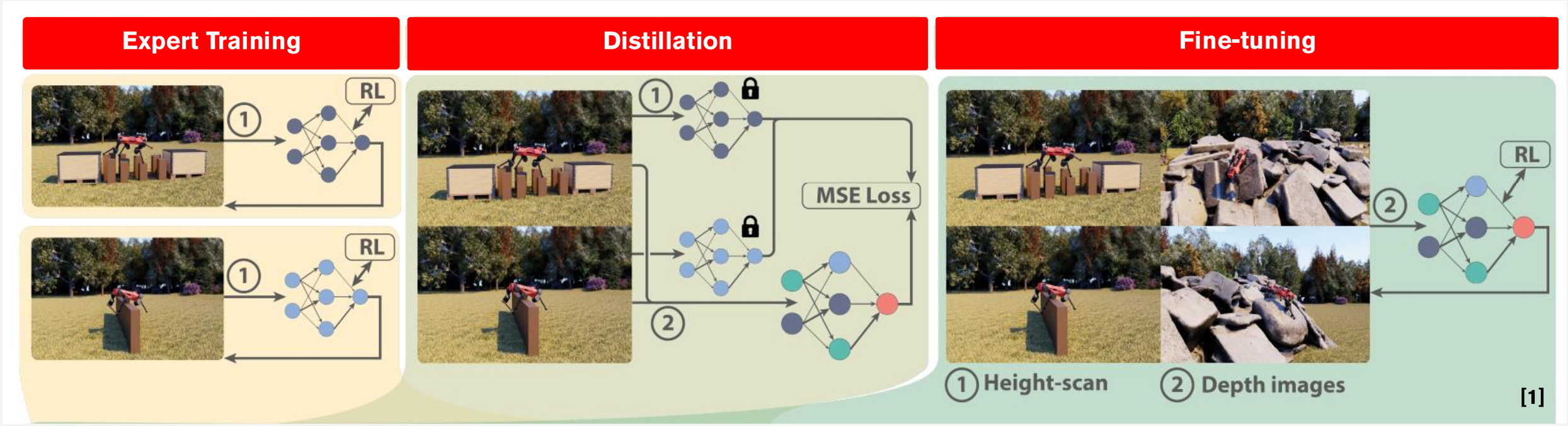
Why ?

After distillation, the student policy is able to imitate all expert behaviors but with lower accuracy and limited generalization capability.

RL from scratch ?

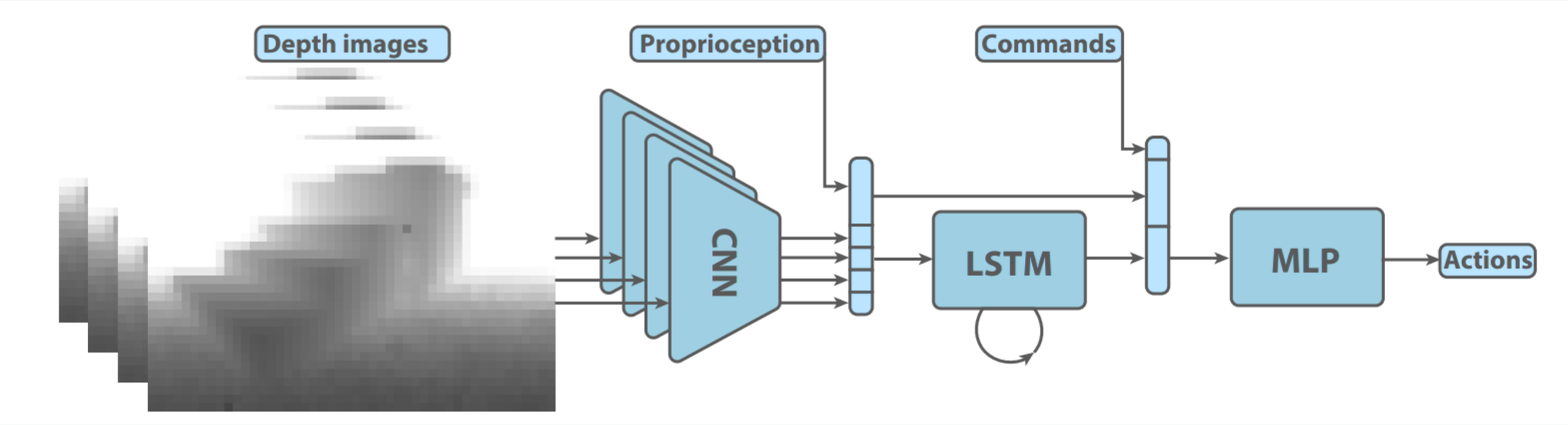
Using RL from scratch would be unstable and inefficient:
The policy only learn a subset of task or terrains.
But starting from the distilled policy provides a strong prior that enables effective learning.

Main structure



Distillation

1/2



(CNN) Convolutional Neural Network
(LSTM) Long Short-Term Memory
(MLP) Multi-Layer Perceptron

Distillation 2/2

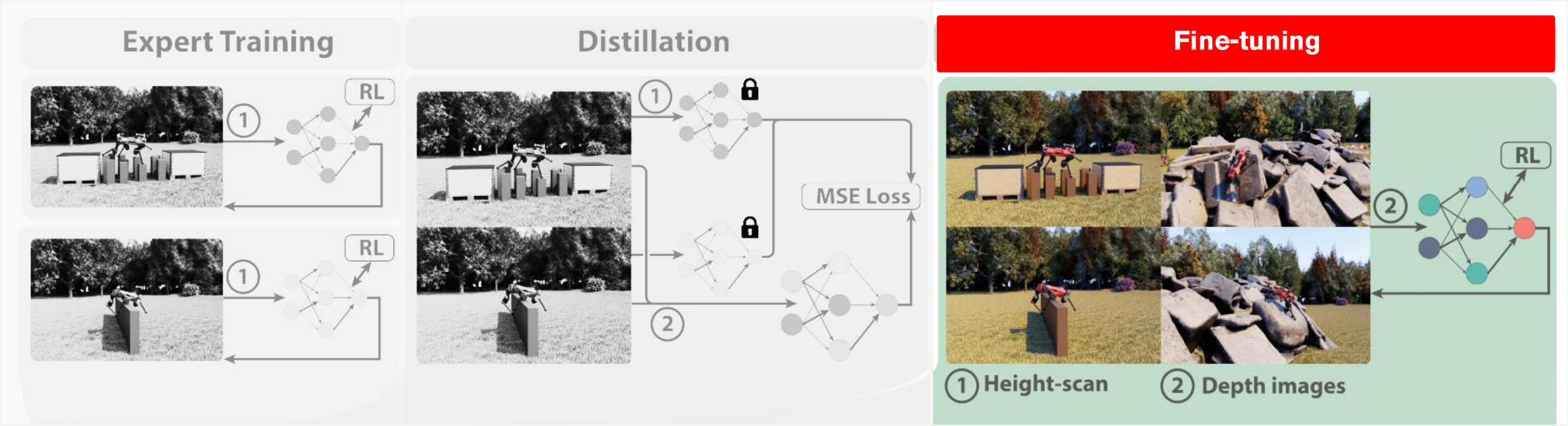
Algorithm 1 Training Scheme for Policy Distillation

$\mathcal{E}_i \leftarrow$ GETEXPERTIDS()
for k in $\leq N_{\text{Epochs}}$ **do**
  $\mathcal{D} \leftarrow \emptyset$
  for $t \leftarrow 0$ to T **do** ▷ Data Collection
    $a_{\text{student}} \leftarrow \pi_{\text{student}}(o_{\text{student}}) + \mathcal{N}(0, \sigma^2)$
    for $e \leftarrow 0$ to N_{experts} **do**
      $a_{\text{expert},\mathcal{I}[\mathcal{E}=e]} \leftarrow \pi_{\text{expert, e}}(o_{\text{expert}})$
    end for
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(o_{\text{student}}, a_{\text{expert}})\}$
    $o_{\text{student},t+1}, o_{\text{expert},t+1} \leftarrow$ ENV.STEP(a_{student})
  end for
  $\pi_{\text{student}} \leftarrow$ TRAIN$(\pi_{\text{student}}, \mathcal{D})$ ▷ Supervised Training
end for

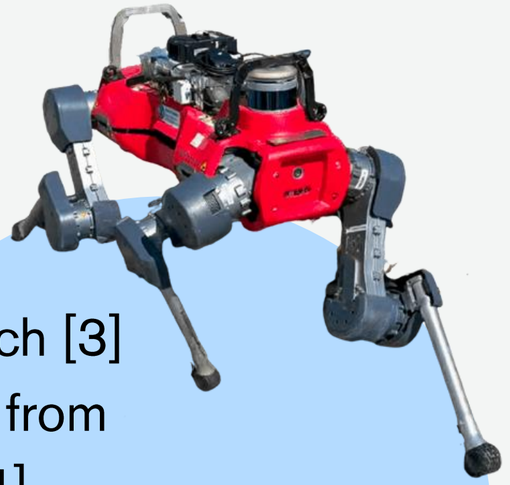
Train section.
This part will be the supervised learning, using the experts.

Fine-tuning

- **maximum performance and robustness** of the foundation policy
- **conservative tuning** of hyper-parameters
- **efficient pre-training** of the critic network



Citations



- (13.09.2025) RSL-RL: A Learning Library for Robotics Research [3]
- (27.08.2025) HERMES: Human-to-Robot Embodied Learning from Multi-Source Motion Data for Mobile Dexterous Manipulation [4]
- (12.08.2025) Large Scale Robotic Material Handling: Learning, Planning, and Control [5].
- (04.08.2025) Learning Autonomous and Safe Quadruped Traversal of Complex Terrains Using Multi-Layer Elevation Maps [6].
- Three additional citations [7, 8 and 9]

Discussion

Pros

- distilled policy extracts **non-trivial knowledge** from the experts
- after fine-tuning, **success rate is 3.1% higher** than corresponding expert
- **better than a hierarchical setup** on complex, unstructured terrain

Cons

- an average of **10.4% drop in success rate** of distilled policy without fine-tuning
- **LSTM** has only short term memory and robots **“forgets” obstacles**
- distilled policy tends to **use knees** more than expert skills as a safer approach

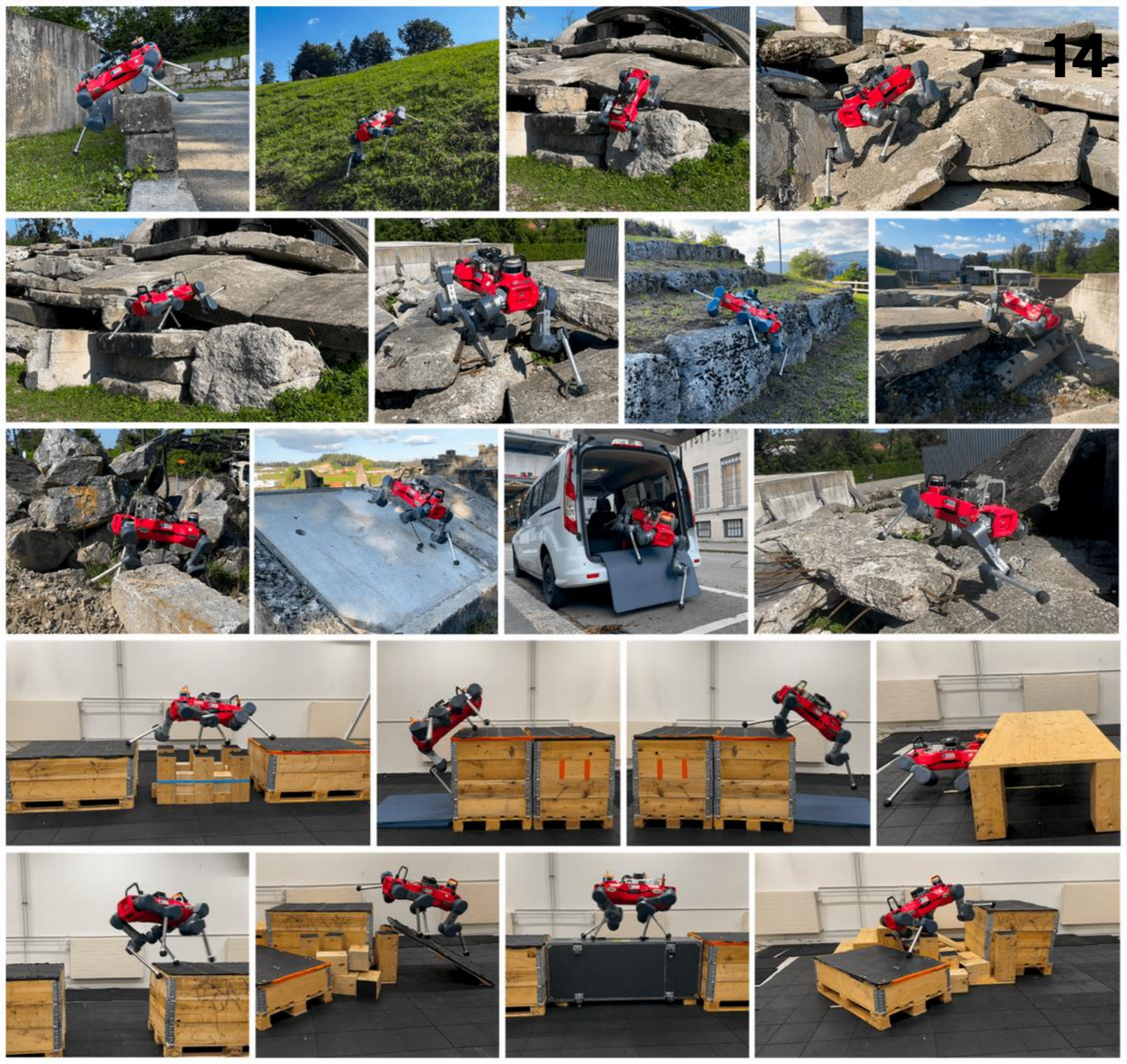
Executive Summary

Robot	ANYmal D (quadruped)
Control	End-to-end torque* control
Design Method	Reinforcement Learning + Distillation (DAgger) + RL fine-tuning.
Gaits / Skills	9 expert skills: walk, climb up, climb down, jump, crouch, low-wall jump, narrow beams, stepping-stones, piles of boulder → 1 single policy
Sensors	Proprioception + 4 Intel RealSense D435i depth cameras

*torque is not mentioned explicitly

Conclusion

Autonomous locomotion requires one policy to tackle every obstacle.



Different scenarios all using the same policy

Joaquin Vasquez-Romo | Paul Huot-Marchand | Josua Bubenzer
Group 12

Parkour in the wild

thank you for your attention

Learning a general and extensible agile locomotion policy using multi-expert distillation and RL Fine-tuning

by Nikita Rudin, Junzhe He, Joshua Aurand, and Marco Hutter

References

[1] Rudin N, He J., Aurand J. and Hutter M (2025) Parkour in the wild: Learning a general and extensible agile locomotion policy using multi-expert distillation and RL Fine-tuning. Available at: https://arxiv.org/abs/2505.11164

[2] Hoeller D, Rudin N, Sako D and Hutter M (2023) Anymal parkour: Learning agile navigation for quadrupedal robots. Available at: https://arxiv.org/abs/2306.14874

[3] Schwarke C., Mittal M., Rudin N., Hoeller D. and Hutter M. (2025) RSL-RL: A Learning Library for Robotics Research. Available at: https://arxiv.org/abs/2509.10771

[4] Yuan Z., Wei T., Gu L., Hua P., Liang T., Chen Y., Xu H. (2025) HERMES: Human-to-Robot Embodied Learning from Multi-Source Motion Data for Mobile Dexterous Manipulation. Available at: https://arxiv.org/abs/2508.20085

[5] Spinelli F. A., Zhai Y., Nan F., Egli P., Nubert J., Bleumer T., Miller L., Hofmann F. & Hutter M. (2025) Large Scale Robotic Material Handling: Learning, Planning, and Control. Available at: https://arxiv.org/abs/2508.09003

[6] Chen Y., Ma J., Luo Z., Han Y., Dong Y., Xu B. & Lu P. (2025) Learning Autonomous and Safe Quadruped Traversal of Complex Terrains Using Multi-Layer Elevation Maps. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=11112615

[7] Sun J., Han G., Sun P., Zhao W., Cao J., Wang J., Guo Y., Zhang Q. (2025) DPL: Depth-only Perceptive Humanoid Locomotion via Realistic Depth Synthesis and Cross-Attention Terrain Reconstruction. Available at: https://arxiv.org/abs/2510.07152

[8] Atanassov V., Yu W., Gangapurwala S., Wilson J. & Havoutis I. (2025) GeCCo — a Generalist Contact-Conditioned Policy for Loco-Manipulation Skills on Legged Robots. Available at: https://arxiv.org/abs/2509.17582

[9] Rudin N. (2025) Generalizing Agile Legged Locomotion with Deep Reinforcement Learning. ETH Zurich Research Collection. [page not found]

Appendix

Algorithm 1 Training Scheme for Policy Distillation

$\mathcal{E}_i \leftarrow$ GETEXPERTIDS()
for k in $\leq N_{\text{Epochs}}$ **do**
  $\mathcal{D} \leftarrow \emptyset$
  for $t \leftarrow 0$ to T **do** ▷ Data Collection
    $a_{\text{student}} \leftarrow \pi_{\text{student}}(o_{\text{student}}) + \mathcal{N}(0, \sigma^2)$
    for $e \leftarrow 0$ to N_{experts} **do**
      $a_{\text{expert},\mathcal{I}[\mathcal{E}=e]} \leftarrow \pi_{\text{expert, e}}(o_{\text{expert}})$
    end for
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(o_{\text{student}}, a_{\text{expert}})\}$
    $o_{\text{student},t+1}, o_{\text{expert},t+1} \leftarrow$ ENV.STEP(a_{student})
  end for
  $\pi_{\text{student}} \leftarrow$ TRAIN$(\pi_{\text{student}}, \mathcal{D})$ ▷ Supervised Training
end for

Table 1. Symbols.

Symbol	Description
$\mathbf{r}$, $\mathbf{r}^*$	Current and target base positions
ψ, ψ^*	Current and target base headings
t^*	Remaining time to reach the target
α	Angle between base z-axis and gravity
$\mathbf{v}_b$, $\boldsymbol{\omega}_b$	Base velocities in base frame
$\mathbf{g}_b$	Gravity vector in base frame
$\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\tau}$	Joint positions and velocities
$\mathbf{q}_{\text{lim}}, \dot{\mathbf{q}}_{\text{lim}}$	Joint limits
$\mathbf{q}^*, \mathbf{q}_d$	Desired and default joint positions
$\boldsymbol{\tau}$, $\boldsymbol{\tau}_{\text{lim}}$	Joint Torques and torque limits
$\mathbf{v}_f$, $\mathbf{F}_f$	Feet linear velocity and contact force
$\mathbf{Em}$	Elevation map around the robot
$\mathbf{Ls}$	Lidar (horizontal) scan around the robot
$\mathbf{I}$	Depth images
$\$_L$	Target reached, $\$_L = \mathbb{1}_{\|\mathbf{r}_{xy} - \mathbf{r}^*_{xy}\| < 0.25} \mathbb{1}_{\|\psi - \psi^*\| < 0.5}$

Table 2. Rewards used during fine-tuning.

Reward Term	Expression	Weight
Track position	$\mathbb{1}_{t^*<1}(1 - 0.5\|\mathbf{r}_{xy} - \mathbf{r}^*_{xy}\|)$	10
Track heading	$\mathbb{1}_{t^*<1}(1 - 0.5\|\psi - \psi^*\|)$	5
Joint velocity	$\|\dot{\mathbf{q}}\|^2$	-1e-3
Torque	$\|\boldsymbol{\tau}\|^2$	-1e-5
Joint vel. limit	$\sum_{i=1}^{12} \max(\|\dot{\mathbf{q}}_i\| - \dot{q}_{\text{lim}}, 0)$	-1
Torque limit	$\sum_{i=1}^{12} \max(\|\boldsymbol{\tau}_i\| - \tau_{\text{lim}}, 0)$	-0.2
Base acc.	$\|\dot{\mathbf{v}}\|^2 + 0.02\|\dot{\boldsymbol{\omega}}\|^2$	-1e-3
Feet acc.	$\sum_{f=1}^{4} \|\dot{\mathbf{v}}_f\|$	-2e-3
Action rate	$\|\mathbf{q}^*_t - \mathbf{q}^*_{t-1}\|^2$	-1e-2
Feet force	$\sum_{f=1}^{4} \max(\|F_f\| - 700, 0)^2$	-1e-5
Don't wait	$\mathbb{1}(\|\mathbf{v}_b\| < 0.2)$	-1
Stand at target	$\$_L \|\mathbf{q} - \mathbf{q}_d\|$	-0.5
Collision	$\mathbb{1}_{\textit{knee/shank collision}}$	-1
Termination	$\mathbb{1}_{\alpha>135^\circ} + \mathbb{1}_{\dot{\mathbf{q}}>\dot{\mathbf{q}}_{\text{lim}}}$	-2e3

Table 3. Observations.

Observation	Expert	Student	Critic
$\mathbf{v}_b$	×	×	×
$\boldsymbol{\omega}_b$	×	×	×
$\mathbf{g}_b$	×	×	×
$\mathbf{q}, \dot{\mathbf{q}}$	×	×	×
$\mathbf{r}^*, t^*, \psi^*$	×	×	×
$\mathbf{Em}_{2\,\text{m}\times 1\,\text{m}}$	×		×
$\mathbf{Em}_{6\,\text{m}\times 3\,\text{m}}$			×
$\mathbf{Ls}_{1ray/30^\circ}$			×
$4 \times \mathbf{I}$		×	

Table 4. Success rate of policies on different terrains. π_w to π_{ss} are the experts trained on *Walk*, *Climb*, *Climb down*, *Jump*, *Tables*, *Rock pile*, *Low wall*, *Beams*, and *Stepping stones*, respectively. π_D and π_{RL} are the distilled and fine-tuned policies. Terrains from *Walk* to *Stepping stones* are used during distillation. *Parkour line* and *Scanned meshes (train)* terrains are added during fine-tuning. The other terrains are never seen during training and are used only to evaluate the generalization capabilities of policies. π_{RL^*} is fine-tuned again on the *Climb down on stones*. The highest success rate across policies and all rates within $0.5\,\%$ are in bold.

Terrain	π_w	π_c	π_{cd}	π_j	π_t	π_{rp}	π_{lw}	π_b	π_{ss}	π_D	π_{RL}	π_{RL^*}
Walk	94.6	54.4	36.7	43.8	23.4	96.4	30.4	25.4	33.2	99.3	**100.0**	**99.8**
Climb	0.0	98.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	98.1	**99.5**	**99.4**
Climb Down	2.6	13.4	**99.9**	47.6	3.1	43.0	2.1	8.8	6.0	84.3	**99.7**	**99.6**
Jump	0.7	2.9	0.0	**98.4**	0.0	9.7	0.0	0.0	0.0	93.5	**98.0**	97.7
Tables	0.2	24.4	0.0	0.0	99.4	37.2	0.0	0.0	0.0	78.9	**100.0**	**100.0**
Rock pile	34.6	14.6	31.5	25.9	2.6	92.2	24.9	4.0	12.4	82.6	96.5	**97.1**
Low wall	0.0	14.8	0.0	0.0	0.0	21.8	84.8	0.0	0.0	77.0	**99.9**	**100.0**
Beams	0.3	3.2	0.0	8.7	0.3	0.4	0.1	97.3	0.3	85.2	**99.5**	**99.5**
Stepping stones	1.1	1.3	0.0	23.4	0.0	0.0	0.0	1.0	**98.8**	73.0	**98.8**	**98.9**
Parkour line	0.2	0.0	0.1	0.2	18.6	17.2	0.0	0.0	0.0	5.8	**98.5**	**98.7**
Scanned meshes (train)	0.2	1.6	0.0	0.0	0.0	44.8	0.0	0.0	0.0	11.9	**99.1**	**98.8**
Scanned meshes (test)	0.2	1.6	0.0	0.0	0.0	44.8	0.0	0.0	0.0	14.9	**94.9**	93.9
Arranged rocks	31.1	55.6	17.2	17.0	6.8	67.7	13.7	13.0	9.7	62.9	**93.2**	**92.8**
Gap - climb	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	10.2	82.0	**86.6**
Down - stones	0.0	0.0	0.0	17.8	0.0	0.0	0.0	0.0	1.3	11.3	54.4	**92.4**

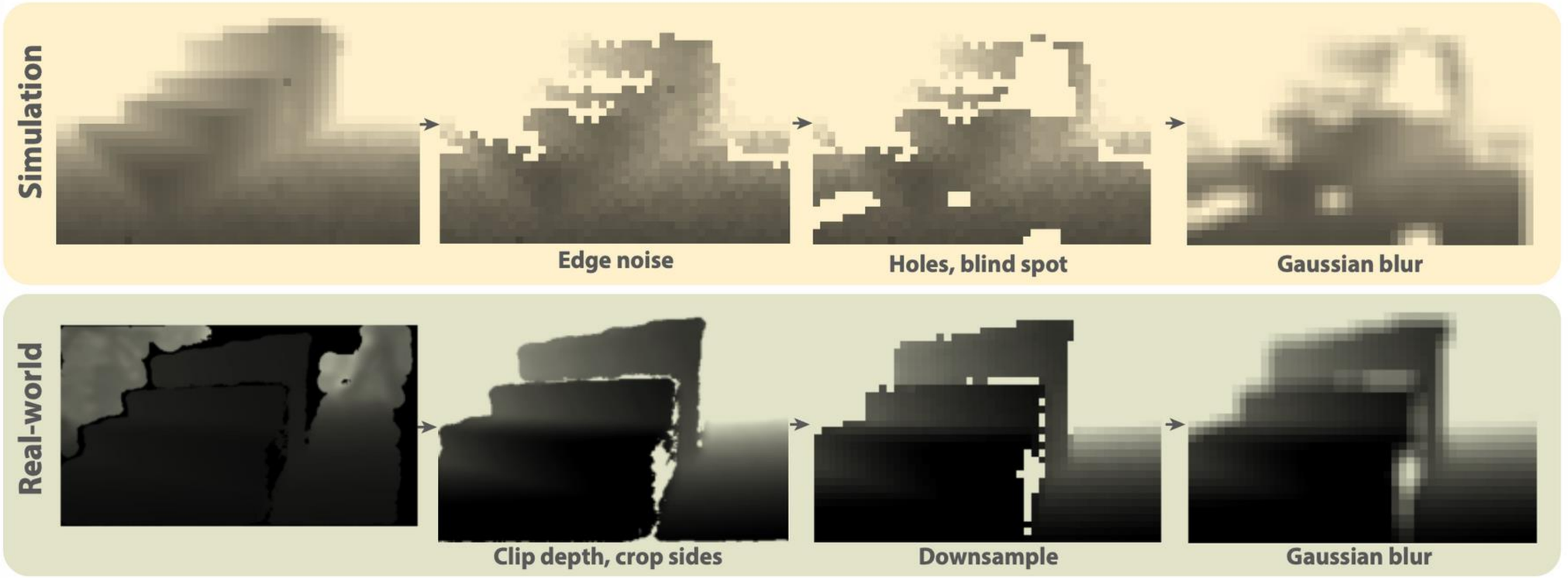


Figure 4. Processing applied to simulated and real depth images. Simulated images are degraded using the following steps: 1) pixels surrounding edges are shuffled and/or removed, 2) random holes are added using slowly evolving Perlin noise, and 3) the image is blurred using a Gaussian filter. Real images are 4) clipped, downsampled, and cropped, and 5) blurred using the same Gaussian filter as in simulation.

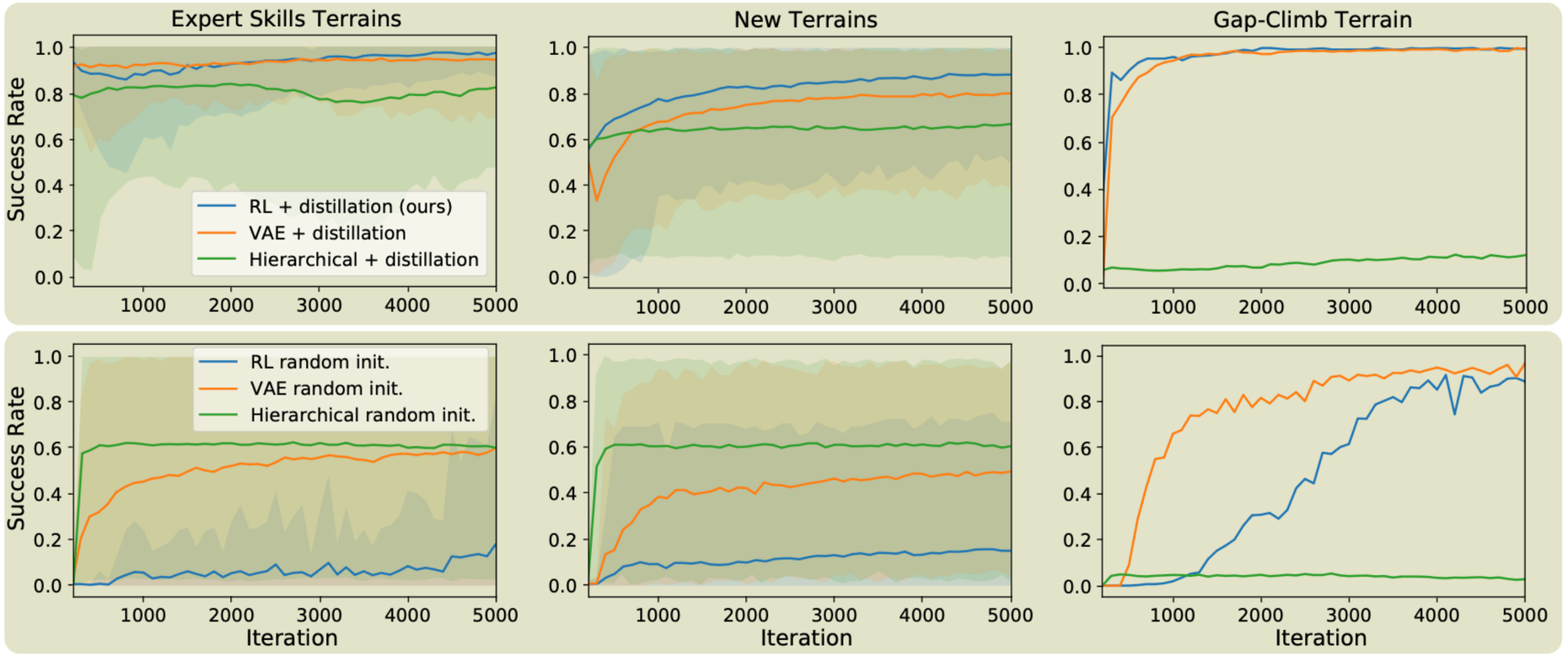
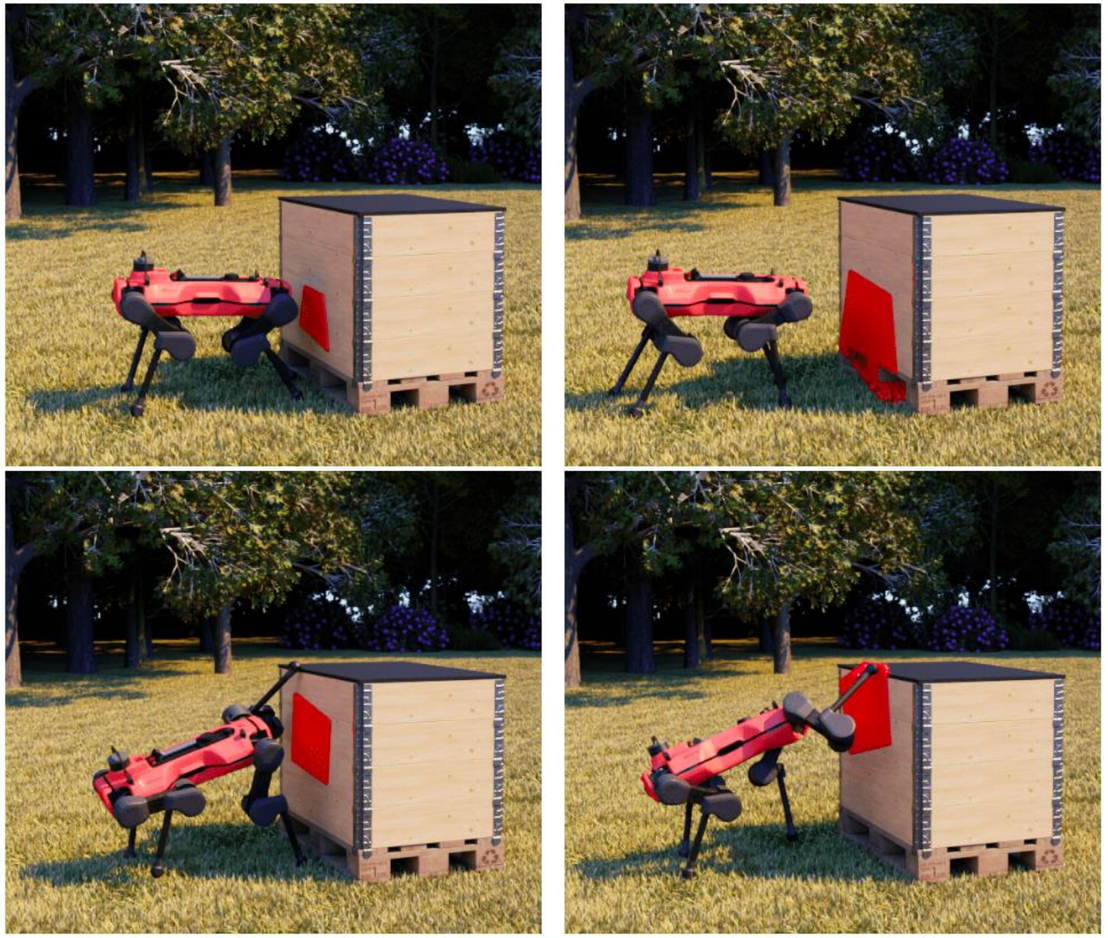


Figure 6. Comparison of three skill combination methods. We compare our proposed distillation + fine-tuning approach to a hierarchical approach where a policy is trained to switch between experts and a method in which we use a VAE to encode motions of skills into a latent space, freeze the decoder and then train a new policy controlling the robot through the resulting latent space. The top row shows the results for pre-trained policies. Both the hierarchical and VAE policies can be trained from scratch or pre-trained in the setup used for distillation by adapting the desired output (one hot encoding and latent vector, respectively). The bottom row shows the results without pre-training (Note that our approach without pre-training is standard RL training from scratch). We evaluate the performance using 1000 robots on 100 randomized terrains with difficulties from $50\,\%$ to $100\,\%$ of the maximum training difficulty. We use the mean and mode of the policies. The solid curve shows the mean success rate across all 100 terrains. The shaded areas show the minimum and maximum success rates across terrain types.



(a) After distillation **(b)** After fine-tuning

Figure 8. Change of behavior between the distilled and fine-tuned policy. After fine-tuning, the policy learns to stay further away from the obstacle to maximize the visibility of the obstacle in the depth camera's field of view.